Rethinking LEO Constellations Routing with the Unsplittable Multi-Commodity Flows Problem

Paul Grislain
ENAC
University of Toulouse
Toulouse, France
paul.grislain@alumni.enac.fr

Nicolas Pelissier
ISAE-SUPAERO
University of Toulouse
Toulouse, France
nicolas.pelissier@student.isae-supaero.fr

François Lamothe
ISAE-SUPAERO
University of Toulouse
Toulouse, France
francois.lamothe@isae-supaero.fr

Oana Hotescu
ISAE-SUPAERO
University of Toulouse
Toulouse, France
oana.hotescu@isae-supaero.fr

Jérôme Lacan
ISAE-SUPAERO
University of Toulouse
Toulouse, France
jerome.lacan@isae-supaero.fr

Emmanuel Lochin
ENAC
University of Toulouse
Toulouse, France
emmanuel.lochin@enac.fr

José Radzik
ISAE-SUPAERO
University of Toulouse
Toulouse, France
jose.radzik@isae-supaero.fr

***Abstract*—This study investigates the performance of an innovative routing protocol inspired by the Unsplittable Multi-Commodity Flow (UMCF) problem. LEO routing schemes are often based on Shortest Path (SP) algorithms, the Floyd-Warshall algorithm is usually chosen to compute these network paths within the constellation and their end-to-end latency. Instead of considering latency as a criterion, we seek to optimize the overall amount of IP traffic crossing the constellation. This criterion can be optimized by considering the Unsplittable Multi Commodity Flow problem associated with the system. To solve this problem, we use a heuristic algorithm based on randomized rounding that was shown to return solutions of good quality of the Unsplittable Multi Commodity Flow problem in the optimization literature. Using network simulation over Telesat constellation, we show this proposal significantly reduces the overall congestion level compared to the standard SP routing schemes.**

***Index Terms*—LEO mega constellations, routing algorithms, Unsplittable Multi-Commodity Flows problem, shorthest path Floyd-Warshal, Hypatia ns3 simulator**

I. INTRODUCTION

Over the past couple of years, ”New Space” [1] companies have attempted to develop a new type of satellite constellations to enable global Internet access, with high bandwidth and low latency. These constellations are typically Low Earth orbit (LEO) mega-constellations consisting of thousands of satellites that aim at extending the Internet coverage to under-served communities worldwide. Some of the largest proposed constellations are SpaceX’s Starlink with 4,425 satellites initially planned [2], which is also the most advanced currently, 2,388 satellites already launched and more than 250,000 subscribers [3], followed by Amazon’s Kuiper with 3,236 satellites planned [4] and Telesat plans of 1,671 satellites [5].

One important goal of LEO constellations is to provide low latency and since the inter-satellite links have the capacity to reach the speed of light in a vacuum, which is higher than in optical fiber used for terrestrial networks, LEO constellations have the potential to compete with terrestrial networks [6]. Operating satellites in low Earth orbits, under 2000 km height above the Earth surface, contributes to reducing round-trip time (RTT) and latency in comparison with satellites in geostationary orbit (GEO) operating at 35,786 km above the Earth surface. However, the advantage of low latency comes to a cost and generates new challenges related to the dynamic of the LEO infrastructure. While GEO satellites can cover wide areas, bringing satellites nearer to Earth considerably reduces their coverage, thus requiring a larger number of satellites for a global cover and leading to a high velocity movement of the satellites on the orbit (each satellite orbits the Earth every $\sim$100 minutes, travelling at $\sim 27,000$ km/h [7]).

In communication networks, the end-to-end latency and link capacity utilization are usually impacted by the routing protocols and the congestion occurring in intermediate routers. These elements can have an important impact on the performance of LEO constellations as well, but due to the huge number of satellites and their high velocity, finding an optimal inter-satellite routing configuration to reduce latency and congestion and to effectively use network capacity is not a simple problem.

Several recent works have focused on studying how the dynamic of mega-constellations may impact the routing and thus the performance of the network. Most of these studies consider routing protocols based on the shortest path to keep low latency [6], [8]–[11].

The most common method to understand the behavior of satellite constellations and their dynamic (*e.g.*, packet-level behavior, topology, routing), as well as to evaluate the end-to-end performance, is simulation. However, evaluation of such networks is a challenging issue since most of existing works use in-house simulators which makes difficult reproduction, testing, and extension of evaluated scenarios and results. To address this issue, the recent work in [10] proposes Hypatia [12], a ns3-based simulation tool, for network analysis of LEO constellations. Hypatia was built for research purpose and allows visualization and packet-level simulation of satellite constellations. The visualization module helps to describe routing, congestion and RTT in a constellation shell. Constellation shells from Starlink, Kuiper and Telesat are considered.

In this paper, we aim at taking advantage of the existing simulation tool, Hypatia and analyze the impact of routing protocols on the end-to-end network performance. The routing protocol used in Hypatia is based on the Floyd-Warshall shortest path algorithm, but following the shortest path, congestion may occur quickly in links connecting densely populated areas. In this paper, our idea is to apply a novel routing protocol based on the Unsplittable Multi-Commodity Flows (UMCF) algorithm described in [13] that could prevent congestion on inter-satellite links by distributing flows on less occupied links. Our intuition is that less congested longer paths may lead to better performance than existing routing protocols based on the shortest path.

Our main contribution in this paper are the following:

- We introduce an innovative routing protocol focusing on maximizing the amount of IP traffic crossing the constellation instead of minimizing the latency. This is considered by using the Sequential Randomized Rounding (SRR) for the Unsplittable Multi-Commodity Flow (UMCF) problem instead of the classical Floyd-Warshall shortest path algorithm;
- To evaluate the performance of the proposed UMCF routing algorithm, we extend previous work on Hypatia ns3-based simulator with the implementation of the UMCF algorithm.
- We experimentally compare UMCF routing with shortest path routing and show that UMCF routing performs better in terms of packet loss for UDP protocol and for several throughput values of the inter-satellite links for the Telesat constellation.

This paper is organized as follows. First, we present related work in Section II, followed by a background on LEO constellations in Section III. Then, we introduce our solving algorithm based on the Sequential Randomized Rounding (SRR) method for Unsplittable Multi-Comodity Flows (UMCF) in Section IV and our approach of integrating the UMCF routing algorithm into the Hypatia simulator in Section V. The routing algorithms are evaluated in Section VI. Finally, Section VII concludes the paper and gives leads for future work.

II. RELATED WORK

Routing over SATCOM might be considered as a low complexity task, considering the determinism that prevails in this type of system. Basically, it is possible to compute the future position of the satellites regarding the various ground relays and to determine the possible connections. Thus, a complete temporal graph of the connections can be built between each network elements. Once this tree is built, it is theoretically possible to find the route between a source and a destination that optimizes a particular function of the chosen metrics. The problem becomes an optimization problem of weight functions over a graph. For instance, the algorithm *Contact Graph Routing* (*CGR*) is based on this optimization result to make its routing decisions [8]. The main complexity relates to the necessary computing time. Actually, the graph of connections previously mentioned can become particularly large, as its size is directly linked to the number of contact opportunities and nodes present in the system. Two possibilities are then offered to improve the computation time of the solution:

- to find a search algorithm on the connections' tree of less complexity;
- to simplify the connection tree without degrading the results, *i.e.,* without removing the optimal route by a clumsy truncation of the graph.

Concerning the first point, the existing algorithms used for tree analysis are already optimal. This leaves the second point, where two cases appear: whether we seek to optimize the delay or the capacity. If we consider only the transmission delay, the goal is to simplify the connection graph both in width, assuming that contacts that are very far in the future will not be useful, and in depth, assuming that the optimal route has a limited number of hops. This idea is presented and developed in H. C. Sanchez's thesis about Store and Forward routing in satellite constellations [14] and has been deeply investigated.

However, if we are interested in throughput rather than delay, there are many mathematical theories and algorithms allowing to find the routing that optimizes the amount of information that can be exchanged between the different nodes of the network. For instance, the max-flox min-cut theorem [15], [16] and the multi-commodity theory [17] which takes up the previous theorem in the case of several flows present in the network to optimize the global flow. As the latter has been only too recently considered, this motivates the present paper that aims at investigating a solving method for the unsplittable flow problem (UFP), that is a widely studied variant of the multi-commodity flow problem. Thus, we will use the Sequential Randomized Rounding (SRR) algorithm described in the paper [13]. The SRR algorithm improves the Randomized Rounding algorithm from [18] by sequentially solving the relaxed (linear) problem, stating some commodities and loop back until finished.

III. BACKGROUND

A. LEO constellations structure

Mega-constellations are composed of hundreds and thousands of satellites that are organized in orbits. Each orbit is characterized by a set of parameters that determine the orbit mechanics, as follows:

- orbit altitude measured from the surface of the Earth. Low Earth orbit satellites operate at an altitude of less than 2,000 km above the Earth surface;
- orbit inclination given by the angle between the orbital plane and the Equator. When the orbit inclination is low, orbits are called equatorial or near equatorial orbits. Orbits that pass above or nearly above the poles on each revolution with inclinations close to 90° are polar orbits.

A set of orbits with the same altitude and inclination crossing the Equator at uniform separation from each

Authorized licensed use limited to: ISAE. Downloaded on January 04,2023 at 13:26:57 UTC from IEEE Xplore. Restrictions apply.

other is called an orbital shell. Recently proposed LEO constellations may have several shells. For instance, Telesat constellation (on which we base our experiments in this paper) consists in two shells with orbit inclination of 98.98°, respectively 50.88° and orbit altitude of 1,015 km, respectively 1,325 km. These two shells will group a total number of 1,671 satellites [19].

B. LEO connectivity

Each satellite can have two types of full duplex communication links: Inter-Satellite links (ISL) that ensure communication between satellites and Ground-to-Satellite links (GSL) for up/down communication with Ground Stations. ISL can be intra-plane connecting neighbor satellites on the same orbit and inter-plane connecting satellites on neighbor orbits. Inter-satellite distances between different planes can vary in time: longest when satellites are positioned over Equator and shortest when satellites are over the poles. Furthermore, if the planes are deployed at different altitudes, the orbital period will be different. Due to the low altitude of orbits, LEO satellites move at rapid speeds ($>$ 27000 km/h) and communication with Ground Stations last only few minutes before handover to another satellite. For this reason, recently proposed LEO constellations are comprising hundreds to tens of thousands of satellites to provide continuous connectivity.

IV. UNSPLITTABLE MULTI-COMMODITY FLOWS AND SEQUENTIAL RANDOMIZED ROUNDING

A. Problem description

The unsplittable flow problem is an extensively studied variant of the classical maximum flow problem. In this problem, one is given a directed or undirected graph, together with capacities on its arcs. A family of commodities, each composed of an origin, a destination, and a demand, is also provided. Each commodity has to route its demand from its origin to its destination through a unique path. The routing must ensure that capacities on the arcs are not exceeded by the flow of the commodities, or at least minimize the violation of the capacities.

This problem is NP-hard which makes it very difficult to find optimal solutions for instances of the problem with several hundred nodes and commodities such as the one we are considering. However, some heuristics and approximation algorithms can find solutions of excellent quality even on such large instances. In this study, we will consider the Sequential Randomized Rounding (SRR) algorithm described in [13] which is an extension of the Randomized Rounding algorithm [18].

B. Mathematical formulation for unsplittable flows

To formally introduce the problem through a mathematical formulation we will use the following notations:

- $G = (V, E)$ is a directed or undirected graph, with V the set of nodes and E the set of arcs
- $L = (o_k, d_k, D_k)_{k \in K}$ is a set of commodities defined by their origin, destination, and demand.
- $(c_e)_{e \in E}$ are capacities on the arcs

We also use the Kronecker notation, δ_x^y equals 1 if $x = y$ and 0 otherwise. The sets of arcs incoming and outgoing of node v will be noted $E^-(v)$ and $E^+(v)$ respectively. With these notations, the unsplittable flow problem can be more formally described with the following mixed integer linear program :

$$\min_{f_{ek}, \Delta_e} \sum_{e \in E} \Delta_e \tag{1a}$$

such that

$$\sum_{e \in E^+(v)} f_{ek} - \sum_{e \in E^-(v)} f_{ek} = \delta_v^{o_k} - \delta_v^{d_k} \qquad \forall k \in K,\ \forall v \in V, \tag{1b}$$

$$\sum_{k \in K} f_{ek} D_k \leq c_e + \Delta_e \qquad \forall e \in E, \tag{1c}$$

$$f_{ek} \in \{0, 1\},\ \Delta_e \in \mathbb{R}^+ \qquad \forall k \in K,\ \forall e \in E. \tag{1d}$$

In this formulation, the variable f_{ek} indicates whether commodity k pushes flow on arc e and the variable Δ_e represents the overflow on arc e. Equation (1b) corresponds to the flow conservation constraints. It ensures that, for each commodity and every node except the origin and the destination of the commodity, the same amount of flow of the commodity goes in and out of the node. Equation (1c) corresponds to the capacity constraints. It ensures that the capacity of an arc is respected or that the overflow is recorded in Δ_e. Finally, the fact that $f_{ek} \in \{0, 1\}$ ensures that each commodity is allowed to use only one path to route its flow.

C. Description of the heuristic

Before diving in the heuristic, we will introduce a central concept for randomized rounding algorithms, which is the linear relaxation of a mixed integer linear program. To obtain the linear relaxation of a mixed integer linear program, one just removes the integrality constraint on the variables that to have one. In our case, this means replacing the constraint $f_{ek} \in \{0, 1\}$ with a constraint $f_{ek} \in [0, 1]$. This relaxation has an impact on the modeled problem, and in our case the linear relaxation of the unsplittable flow problem is the multi-commodity flow problem. This relaxed problem is identical to the previous one, except that each commodity is allowed to split its flow on several paths.

Let us now describe the heuristic. The SRR algorithm, presented in Algorithm 1 alternates between two different steps:

- solving the linear relaxation where each commodity can send its flow on several paths
- fixing some commodities to a unique path among those proposed by the linear relaxation using randomized rounding.

These steps are iterated upon until all the commodities have been assigned to a unique path. Solving the linear relaxation is easier than solving the original problem because polynomial time algorithms are known to efficiently solve the multi-commodity flow problem. This can be done, for example, by the above formulation to a linear programming commercial solver, but more specific algorithms have been designed in the multi-commodity flow literature [20], [21]. In our experiments, we use the commercial solver Gurobi [22] on the above formulation with an aggregation technique presented in [13] to reduce the computing time. More details on when to actualize the linear relaxation between randomized rounding steps are also given in [13].

As for the fixing of the path of a commodity, a randomized rounding step proceeds as follows. In the solution of the linear relaxation, each commodity may send its flow on several paths. Let us note x_{pk} the proportion of the flow of a commodity k that is sent on path p. A path will be chosen randomly among the one used by commodity k in the linear relaxation. The probability of choosing path p is x_{pk}. The commodity is then forced to use only this path in the final solution and in the subsequent resolution to the linear relaxation.

An important detail of the SRR algorithm is the order in which the commodities are fixed to a unique path. In the SRR heuristic, paths are assigned to the commodities in decreasing order of commodity's demand, *i.e.,* the commodities with larger demands have their paths chosen first. The rationale behind this ordering is to allocate commodities with a large demand first, while a large amount of capacity is left in the arcs. Commodities with smaller demands are then used to fill the remaining gaps. It has been shown that this order has a large impact on the quality of the solution returned by the heuristic.

Algorithm 1 The SRR heuristic

```
Require: $G = (V, E, c)$ a capacitated graph, $L = (o_k, d_k, D_k)_{k \in K}$ a list of commodities
Sort the commodities by decreasing demand
Set $K_{fixed} = \varnothing$    ▷ $K_{fixed}$ is the set of indices of commodities fixed to a single path
for each commodity $k^*$ in decreasing demand order do
    if an actualization is needed then
        $((x_{pk})_{p \in P_k})_{k \in K} = Solve\_Linear\_Relaxation\left(G, L, K_{fixed}, (p_k)_{k \in K_{fixed}}\right)$
    Draw a path $p^*$ from $P_{k^*}$ with probability $x_{p^* k^*}$
    Add index $k^*$ to $K_{fixed}$.
    $p_{k^*} = p^*$
return $(p_k)_{k \in K}$
```

D. On the constellation dynamic

Because they orbit the earth at low altitude, constellations of satellites are highly dynamic systems. To take this characteristic into account, the constellation is thus considered at different time steps. Moreover, due to the ever-changing structure of the network in these systems, users need to change the path they use to transmit their traffic over time. However, path changes disrupt the connection of the user and tend to lower the quality of service. To have a high service quality, the routing decisions must try to minimize the number of path changes over time. Thus, when making routing decisions at a time step, we seek to penalize routing decisions that do not use the same path as in the previous time step. This can be done by slightly modifying the UMCF problem and thus the mixed integer linear program that models it. More precisely, if we note x_k the variable deciding whether user k uses the path p_k from the previous time step and P the penalization endured in case of a path change for any user, then the modified formulation for the UMCF problem is the following:

$$\min_{f_{ek}, \Delta_e} \sum_{e \in E} \Delta_e + P \sum_{k \in K} (1 - x_k) \tag{2a}$$

such that

$$\sum_{e \in E^+(v)} f_{ek} - \sum_{e \in E^-(v)} f_{ek} = \delta_v^{o_k} - \delta_v^{d_k} \qquad \forall k \in K,\ \forall v \in V, \tag{2b}$$

$$\sum_{k \in K} f_{ek} D_k \le c_e + \Delta_e \qquad \forall e \in E, \tag{2c}$$

$$x_k \le f_{ek} \qquad \forall e \in p_k, \forall k \in K \tag{2d}$$

$$f_{ek} \in \{0, 1\},\ \Delta_e \in \mathbb{R}^+ \qquad \forall k \in K,\ \forall e \in E. \tag{2e}$$

The new term $P \sum_{k \in K} (1 - x_k)$ in the objective function counts the number of penalization induces by the solution, and the new constraint $x_k \le f_{ek}$ links the variable x_k with the variables f_{ek} that decide what path user k uses.

V. INTEGRATION OF THE UMCF ROUTING ALGORITHM IN THE SIMULATION FRAMEWORK

A. The simulation environment

Hypatia contains a set of tools to study satellite networks. We enumerate hereafter the main ones:

- **Gather constellation shell information and create a graph**. The graph typically contains Amazon's Kuiper closest shell of the constellation and a set of ground stations. To make simulation more realistic, we can use ground stations located in the 100 most crowded cities.
- **Generate Shortest Path routing**. At regular time steps, node positions and links are updated, and the shortest path between two ground stations is computed. We get routing tables which are ready to use in next simulations. One may notice that a graph contains two static subparts:
  - the satellite graph. One satellite is connected to the same four satellites of the constellation during the whole simulation;
  - the ground stations graph.

The satellites rotate around Earth, and the route generator sets up ground-satellite links and updates satellite-satellite distances.

- **Configure simulation**: choose commodities (from city, to city, data rate or data size), configure links capacities, congestion control (TCP or UDP), data to record (mainly throughput and latency) and other specific parameters.
- The ns3 simulator uses these parameters and the pre-computed routing tables to simulate the data flows in the network.
- Post-processing tools are used. Measures can be compared to theoretical analysis based on the routing tables.

Different configurations of satellites and ground stations interfaces can be used. We mainly used the configuration where a ground station can only connect to one satellite, but a satellite can be connected to many ground stations via the same ground interface embedded in the satellite. Connection between a ground station and a satellite is a complex problem that can have important impacts of the choice on the routing aspects. So we decided to simplify this aspect and focus on network routing by considering that the ground station is connected to the nearest satellite.

Hypatia brings very simple laser inter-satellite connection, and radio connection to the ground.

To test the behavior of the UMCF routing algorithm, we have implemented it into the Gurobi solver, and we have integrated it into Hypatia with the relevant input parameters

B. Integration of the UMCF routing algorithm in Hypatia

We customized Hypatia to simulate the network with UMCF routing algorithm. The process is represented in figure 1.

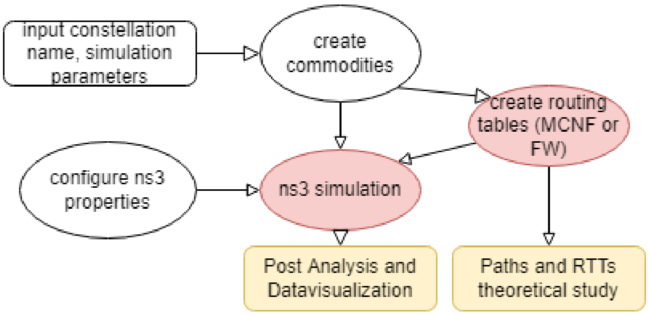

Fig. 1: flow diagram of hypatia scripts

The Unsplittable Multi-Commodity Flows (UMCF) and the Floyd-Warshall (FW) algorithms differ a lot in the way to solve the problem. As input, FW algorithms uses a graph weighted with the distance of each link and outputs directly the routing table, but UMCF needs also commodities, and outputs their optimal paths. We implement interface functions to transform input graph and output results adapted to UMCF. For UMCF, the inputs are the weighted graph with links bandwidths and the list of commodities. The output is the list of paths for each commodity. The interface has to handle exception cases as well, where a ground station cannot connect to any satellite, raising an error in the UMCF implementation. Finally, Hypatia allows running simulations at a packet-level with ns3. The results and plots generated via simulation may be visualized with 3D plots are obtained through CesiumIon [10].

VI. PERFORMANCE EVALUATION BY SIMULATION

A. Experimental setup

The purpose of this experiment is to show the impact of the routing algorithm. To assess its performance on the capacity links and the distribution of the flows in the network, we use UDP traffic. Regarding graph generations, we base our simulation on the first shell of the Telesat constellation. We consider this constellation (351 satellites at 1,015 km, distributed over 27 orbits, at an inclination of 98.98° [19]) because it has fewer satellites than its competitors, so simulations require less calculation resources. Figure 2 presents a 3D view of this constellation, generated with CesiumION in Hypatia. As it can be seen, each satellite has 4 full-duplex links with other satellites: 2 links connecting the satellite with neighbors in the same orbit, and 2 links allowing communication with other satellites in adjacent orbits.

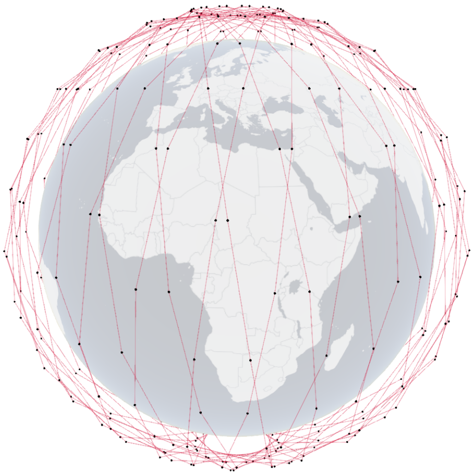

Fig. 2: 3D representation of the first shell of Telesat

The model of communications between ground stations and satellites is very simple: each ground station has one interface to communicate with the satellite and each satellite has one ground interface allowing communication with several ground stations. The ground interface of the satellite is limited by a maximum capacity, *e.g.*, the same as the ISLs. GSLs are also full-duplex.

In our experiments, we consider one ground station in each 100 largest cities of the world. Each ground station is connected to the nearest satellite. Communication flows are generated by the commodity generator as follows. First, it creates 50 pairs between the 100 most populated cities in the world by a random combination without repetition. Then, for each of these 50 couples, UDP communication flows are exchanged between connected cities at a random data rate value ranging from 0.7 to 1.3 Mbps.

We aim at studying how the routing protocols perform when congestion occurs, so we introduce this phenomenon by varying the capacity of Inter-satellite links (ISL) and Ground to Satellite links (GSL). In our experiments, we thus consider the links capacity ranging from 2 Mbps (highly congested links) to 10 Mbps (nearly no congestion).

In our experiments, we consider scenarios of 26 seconds, where routing tables are updated every 2 seconds. The distance computations between nodes are updated much more often to create a continuous satellite movement. At the end of an ns3 simulation, we collect the number and the size of the sent and received packets.

B. Simulation results

In Figure 3, we evaluate the throughput of the routing algorithms for different ISL capacities. This throughput is a simple ratio between the total amount of data received over data sent. When the ISL link capacity is high, near 10 Mbps, we observe no congestion and both algorithms get similar results. But as the ISL capacity decreases, there is more congestion and we observe that distributing flows on different routes reduces losses.

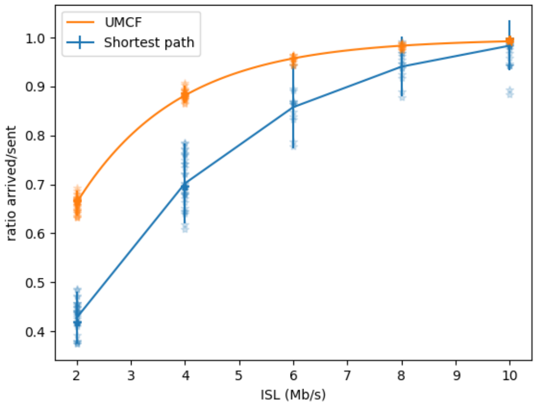

Fig. 3: Ratio between network throughput and link flow for the different algorithms

In Figure 4, we evaluate the differences between cities. We get the commodities throughput of every simulation. By aggregating them by the source city, we get around 15 measurements per city. For each city, we take the median value of these measurements for different ISL rates. The commodities differ a lot by their destination and their flow demand. The standard deviation between results is large, but taking the median value gives us an idea of a representative travel, with a good compromise between the data path length and the flow limitations.

When the ISL rate is high, there are few differences between cities. This can be checked by seeing that success ratio is near of 1 for all scenarios involving an increasing number of cities on the X-axis of Figure 4.

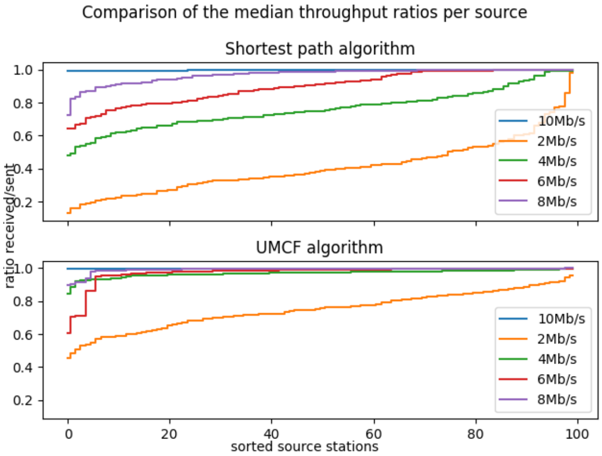

Fig. 4: Median success rate of commodities gathered by departure city

When the ISL rate decreases, we observe that this median value varies much more. Using the same commodities, results show that depending on the routing algorithm, the commodities are not served fairly. With the Shortest Path routing algorithm, we see that a few cities get very good results at the expense of all the other commodities. On the contrary, the UMCF routing algorithm helps to flatten this plot, the measured packet ratio becoming closer to 1 for low ISL data rates.

The results of the routing table is, as expected, different for the two algorithms. Figure 5 shows that Shortest Path algorithm opts for the northern route where distance between satellites is shorter. The UMCF routing algorithm takes a more equatorial route with the same number of satellites, and avoids congestion. Actually, the proposed UMCF algorithm penalizes link utilization, which involves in a smaller number of nodes. However, it does not penalize total distance. Adding penalization on distance is a good way to reduce latency (we observe northern paths), but does not improve significantly the throughput.

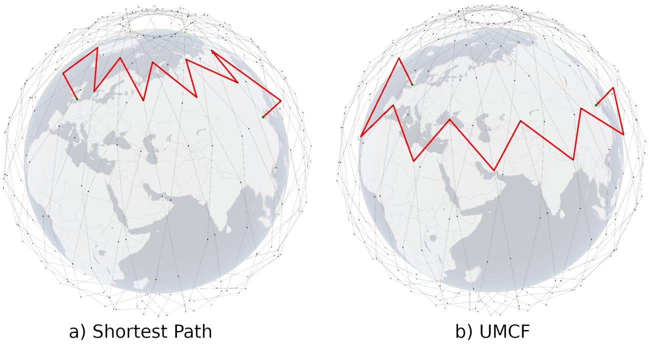
a) Shortest Path b) UMCF

Fig. 5: 3D representation of the routing path calculated with FW and UMCF algorithms from Tianjin to London

In Figures 6, 7, 8 and 9, we study flow distribution around Earth on Inter Satellite links. Big red links are near or above congestion, and tiny green links are underused. Unused links are not represented. In Figures 6 and 7, 10 Mbps ISL capacity is considered. Since this capacity is high enough to serve all commodities in the configuration, the main difference that can be observed is a near-or-above congestion use of polar ISLs.

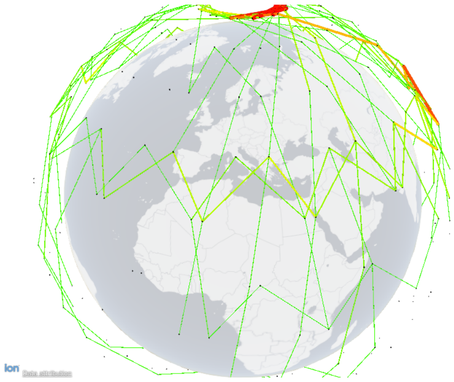

Fig. 6: Example of flow distribution around Earth with Shortest Path routing algorithm and 10 Mbps ISL. Red links show congestion.

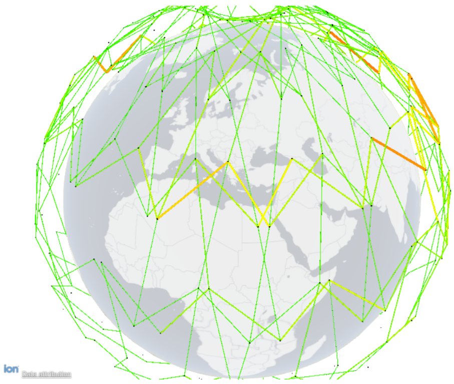

Fig. 7: Example of flow distribution around Earth with UMCF routing algorithm and 10 Mbps ISL. Red links show congestion.

In Figures 8 and 9, we lower the data rate of ISLs to 4 Mbps, to test a congestion behavior. However, it has to be mentioned that red links on these figures only mark congestion, and do not show whether the link capacity is (nearly) fully used or whether the link capacity is exceeded. To compare the performance of the two routing algorithms in terms of congestion, we can observe the mesh of generated routing paths. One can observe that the Shortest Path algorithm in Figure 8 uses less links than the UMCF algorithm in Figure 9. This leads to a lot of congestion on a few overused links, for instance over the North Pole and thus a reduced throughput when the network is not oversized. As it can be noticed, the UMCF routing algorithm tries to take advantage of the maximum link capacity when it is possible, for which reason several links may appear in red and still under congestion. We can thus conclude that an overall better distribution of the flows in the constellation is obtained with the UMCF algorithm, as expected.

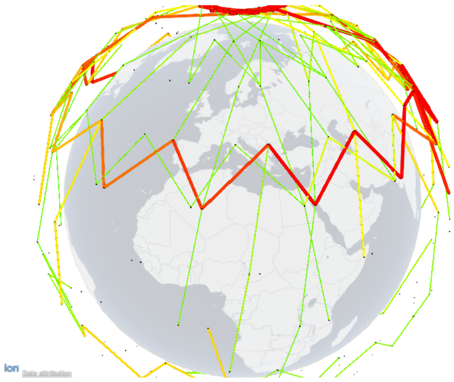

Fig. 8: Example of flow distribution around Earth with Shortest Path routing algorithm and 4 Mbps ISL. Red links show congestion.

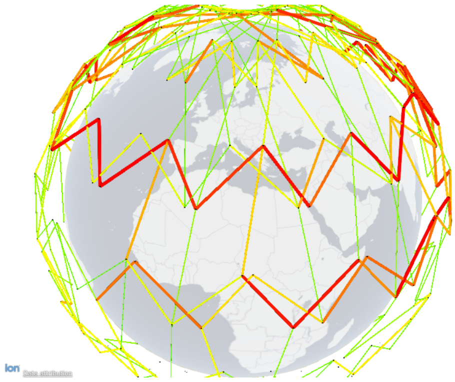

Fig. 9: Example of flow distribution around Earth with UMCF routing algorithm and 4Mb/s ISL. Red links show congestion.

VII. CONCLUSION

In this study, we focus on the performance of an Unsplittable Multi-Commodity Flow (UMCF) algorithm to generate routing tables on the Telesat LEO mega-constellation. To simulate the performance over UDP protocol, we integrate the UMCF algorithm in Hypatia ns3-based simulator.

We show that for high ISL data rates, when there is no congestion in the network, *i.e.*, the network is oversized, the shortest path computed with Floyd-Warshall algorithm and the UMCF algorithm have similar performance. But when congestion occurs, results show that the proposed UMCF routing algorithm establishes better routing tables than shorthest path algorithm for UDP traffic.

As future work, we envision to extend our experiments to cover the case of data transmissions with the TCP protocol. We aim at simulating different versions of the protocol and investigate the impact of the routing protocol on the Quality of Service of applications. The impact of allowing the ground station to choose the first satellite of the path among all its visible satellites will also be studied.

REFERENCES

[1] “NewSpace: The ”Emerging” Commercial Space Industry,” https://ntrs.nasa.gov/.../20140011156.pdf, accessed: 2022-04-27.

[2] “Fcc authorizes spacex to provide broadband satellite services,” https://www.fcc.gov/document/fcc-authorizes-spacex-provide-broadband-satellite-services, accessed: 2022-04-27.

[3] “Spacex’s approach to space sustainability and safety,” https://www.spacex.com/updates/, accessed: 2022-04-27.

[4] “Amazon receives FCC approval for Project Kuiper satellite constellation,” https://www.aboutamazon.com/news/company-news/amazon-receives-fcc-approval-for-project-kuiper-satellite-constellation, accessed: 2022-04-27.

[5] “Telesat lightspeed technology infographic,” https://www.telesat.com/wp-content/uploads/2021/04/Telesat-Lightspeed-Technology-Infographic.pdf, accessed: 2022-04-27.

[6] M. Handley, “Delay is not an option: Low latency routing in space,” in *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, 2018, pp. 85–91.

[7] D. Bhattacherjee and A. Singla, “Network topology design at 27,000 km/hour,” in *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, 2019, pp. 341–354.

[8] J. A. Fraire, O. De Jonckere, and S. C. Burleigh, “Routing in the space internet: a contact graph routing tutorial,” *Journal of Network and Computer Applications*, vol. 174, p. 102884, 2021.

[9] M. Handley, “Using ground relays for low-latency wide-area routing in megaconstellations,” in *Proceedings of the 18th ACM Workshop on Hot Topics in Networks*, 2019, pp. 125–132.

[10] S. Kassing, D. Bhattacherjee, A. B. Águas, J. E. Saethre, and A. Singla, “Exploring the” internet from space” with hypatia,” in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 214–229.

[11] A. Valentine and G. Parisis, “Developing and experimenting with leo satellite constellations in omnet++,” *arXiv preprint arXiv:2109.12046*, 2021.

[12] snkas, “Github of hypatia,” accessed: 2022-03-10. [Online]. Available: https://github.com/snkas/hypatia

[13] F. Lamothe, E. Rachelson, A. Haït, C. Baudoin, and J.-B. Dupé, “Randomized rounding algorithms for large scale unsplittable flow problems,” *Journal of Heuristics*, vol. 27, no. 6, pp. 1081–1110, 2021.

[14] H. Cruz Sanchez, “Routage pour une constellation de satellites store and forward dans des conditions difficiles,” Ph.D. dissertation, Paris, ENST, 2008.

[15] T. Leighton and S. Rao, “Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms,” *Journal of the ACM (JACM)*, vol. 46, no. 6, pp. 787–832, 1999.

[16] G. B. Dantzig and D. R. Fulkerson, “On the max flow min cut theorem of networks,” Rand Corp Santa Monica, Ca, Tech. Rep., 1955.

[17] T. Leighton and S. Rao, “An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms,” Massachusetts Inst Of Tech Cambridge Microsystems Research Center, Tech. Rep., 1989.

[18] P. Raghavan and C. D. Tompson, “Randomized rounding: a technique for provably good algorithms and algorithmic proofs,” *Combinatorica*, vol. 7, no. 4, pp. 365–374, 1987.

[19] Telesat, “Application for modification of market access authorization,” 2020, accessed: 2022-03-10. [Online]. Available: https://fcc.report/IBFS/SAT-MPL-20200526-00053/2378318.pdf

[20] L. K. Fleischer, “Approximating fractional multicommodity flow independent of the number of commodities,” *SIAM Journal on Discrete Mathematics*, vol. 13, no. 4, pp. 505–520, 2000.

[21] W. Dai, J. Zhang, and X. Sun, “On solving multi-commodity flow problems: An experimental evaluation,” *Chinese Journal of Aeronautics*, vol. 30, no. 4, pp. 1481–1492, 2017.

[22] L. Gurobi Optimization, “Gurobi optimizer reference manual,” 2020. [Online]. Available: http://www.gurobi.com